Chapter 4

Color Image Watermarking With Dual RST Attacks
CHAPTER 4
COLOR IMAGE WATERMARKING WITH DUAL RST ATTACKS

4.1 INTRODUCTION

In the universe of communication, security assumes a basic part and claims a major administration looking into its stack. Subsequently security turns with make the way with open a correspondence box (Aparna and Ayyappan, 2014). From our Literature It will be Prove that No watermarking Algorithm is there for preventing the Geometric attacks (Lin, Tiegang, Sheng, And Zhang, 2014). We have made framework to do secure exchange which is visual cryptography conspire and, for copyright assurance and manage geometrical attacks the watermarking plan is utilized. It's totally unthinkable that anyone could disentangle the information contained inside some of offers.

4.2 COLOR IMAGE WATERMARKING BLOCK SCHEMATIC

In the wake of concentrate different visual cryptography system and watermarking strategies, we propose new procedure for secure bank exchange. In this system we give validness and information respectability of the offers utilizing watermark system. In our plan we take one QR-Code as unique image or host image and make shares utilizing 2-out-of-2 VC plot (Shyong and Ming Chiang Chen, 2015). At the point when two offers will be made, server share is put away in bank database and customer share is kept by client. The client will give customer share amid every one of the exchanges with bank. After that we apply the watermark procedure on that customer share image for giving the validation and information respectability and send it on the open correspondence channel.

Figure 4.1: Proposed Block Diagram for Color Image
A. QR-Generation: First select the client name and secret word. Presently utilizing Zxing library creating the QR-code. That QR-code is presently in imperceptible shape so now one can see the information inside. Promote we have Apply VCS plan to produce two offers of QR-Code.

B. Embedding: In this procedure select the shading spread image. Concentrate the R, G and B part. Presently Select R-segment and Apply P-Zemike Moment and DWT-SVD change and Extract LL-bit. In the LL-Bit inserting the Share-1 information. After Invers DWT-SVD change to produce R-Embedded Image Now Add Remain G and B Component to Create Color Water Mark Image. Shading Watermark Image is transmitted over the Network Different Attackers Apply RST assaults on it.

C. Extraction: After RST assaults getting the Attack Color Image Which is currently apply the P-Zemike Moment with Surf Feature Extraction to recuperate assaults. Presently Extracting the offer 1 and it will consolidate with another database share 2 to produce QR-Code. QR decoder will disentangle the Username and Password.

The magnificence of our framework lies in the way that, if any aggressor influences a duplicate of any image to share to produce it later, the watermark will be mutilated so for such manufactured share. Our framework won't permit the age of host image from the heap of 2 shares. In this way, the aggressor won't get the first image.

Here we utilize Singular Value Decomposition discrete wavelet change based watermarking strategy which is geometrically invariant. This write watermarking plan is hearty against the RST assaults, different JPEG and clamor assaults.

4.3 ENCODER ALGORITHM
Step 1: Enter User name and Password.
Step 2: Encode to QR-Image.
Step 3: Apply VCS and Generate 2-Share.
Step 4: Share 2 is Save in Database.
Step 5: Select Color Cover Image.
Step 6: Extract R-Component.
Step 7: Apply Block DWT + SVD + Pseudo Zemike Moment.
Step 8: Embedding Share 1 in LL-band, G and B to Generate Watermark image.
Step 9: Apply Rotation, Scale and Translation on Watermark Image.

4.4 DECODER ALGORITHM
Step 1: Read Attack Watermark Image.
Step 2: Extract R-Component.
Step 3: Apply Pseudo Zemike Moment.
Step 4: Apply Surf Feature Extraction and Affine Transformation.
Step 5: Recover Rotation, Scale and translation Attacks.
Step 6: Apply Block DWT + SVD.
Step 7: Extract Share1 from LL-band.
Step 8: Combine Share 1 and Share 2.
Step 9: Decode QR-Image.
Step 10: Recover User name and Password.

4.5 EXPERIMENTAL RESULTS

4.5.1 QR-CODE GENERATION

![QR-Code generation](image)

a. Fill ID and Password  

b. Generated QR-Code

Figure 4.2: QR-Code generation

As shown in the Figure 4.2 the data is enter by user and QR-Code is generated according to the data.

4.5.2 2-VISUAL CRYPTOGRAPHY

![Share-1 and Share-2 Generation](image)

a. Share-1  
b. Share-2

Figure 4.3: Share-1 and Share-2 Generation

As shown in the Figure 4.3 QR-Code separated by the visual cryptography algorithm in two shares. Share-1 is use for Embedding and Share-2 is save at the receiver side safely.
4.5.3 DWT-SVD WATERMARKING

![Cover image](image1)

![DWT of image](image2)

![Watermark Image](image3)

**Figure 4.4: DWT-SVD**

As shown in the Figure 4.4 cover image is further apply DWT-SVD so I got the LL-band and that is use for data embedding and finally I got watermarked image as shown in Figure c.

4.5.4 ROTATION ATTACKS

![Apply 30° attack](image4)

![Recover Rotational attacks](image5)
Figure 4.5: Rotation Attacks

As shown in the Figure 4.5 the first Figure a shows the attack apply to the watermark image at the angle of 30°. Now in b, using surf feature and affine transformation the angle is recovered near by the attack angle. As shown in c. I applied the affine transformation to recover the watermark image rotational angle. So by applying extraction I got the share-1 as shown in d. at the end by combining the share-1 and share-2 will get the exact QR-code image as shown in e. and calculating the PSNR and MSE value as shown in f.
4.5.5 SCALING ATTACKS

![Scaling Attacks Image]

Figure 4.6: Scaling Attacks

As shown in the Figure 4.6 the first figure a shows the attack apply to the watermark image at the Scale of 2. Now shown in b, using surf feature and affine transformation the Scaling is recovered near by the attack scale. As shown in c, I applied the affine transformation to recover the watermark image rotational angle. So by applying extraction I got the share-1 as shown in d. at the end by combining the share-1 and share-2 will get the exact QR-code image as shown in e. and calculating the PSNR and MSE value as shown in f.
4.5.6 TRANSLATION ATTACKS

As shown in the Figure 4.7 the first Figure a shows the Translation attack apply to the watermark image at the value of Translation 10. Now shown in b, using surf feature and affine transformation the Translation is recovered near by the attack. As shown in c. I applied the affine transformation to recover the watermark image Translation. So by applying extraction I got the share-1 as shown in d. at the end by combining the share-1 and share-2 will get the exact QR-code image as shown in e. and calculating the PSNR and MSE value as shown in f.
4.5.7 ROTATION-SCALE ATTACKS

As shown in the Figure 4.8 the first Figure a shows the rotation-scaling attack apply to the watermark image at the value of rotational angle $30^\circ$ and scale 2. Now shown in b, using surf feature and affine transformation the rotation-scaling is recovered near by the attack rotation-scaling. As shown in c, I applied the affine transformation to recover the watermark image rotation-scaling. So by applying extraction I got the share-1 as shown in d, at the end by combining the share-1 and share-2 will get the exact QR-code image as shown in e, and calculating the PSNR and MSE value as shown in f.

Figure 4.8: Rotation-Scale Attacks
4.5.8 ROTATION-TRANSLATION ATTACKS

![Image of rotation-translation attacks with examples](image)

- a. R-30° T-10 Attack
- b. Recover RT
- c. Recover image
- d. Extracted Share 1
- e. Decode QR-code
- f. PSNR and MSE

**Figure 4.9:** Rotation-Translation Attacks

As shown in the Figure 4.9 the first Figure a shows the rotation-translation attack apply to the watermark image at the value of rotation angle 30° and translation 10. Now shown in b, using surf feature and affine transformation the rotation-translation is recovered near by the attack rotation-translation. As shown in c, I applied the affine transformation to recover the watermark image rotation-translation. So by applying extraction I got the share-1 as shown in d. at the end by combining the share-1 and share-2 will get the exact QR-code image as shown in e. and calculating the PSNR and MSE value as shown in f.
4.5.9 **TRANSLATION-SCALE ATTACKS**

![Figure 4.10: Translation-Scale Attacks](image)

As shown in the Figure 4.10 the first Figure a shows the translation-scale attack apply to the watermark image at the value of Translation 10 and Scaling 2. Now shown in b, using surf feature and affine transformation the translation-scale is recovered near by the attack translation-scale. As shown in c. I applied the affine transformation to recover the watermark image translation-scale. So by applying extraction we get the share-1 as shown in d. at the end by combining the share-1 and share-2 will get the exact QR-code image as shown in e. and calculating the PSNR and MSE value as shown in f.
4.5.10 ROTATION, SCALE AND TRANSLATION ATTACKS

![Image of RST Attack](image1)

a. RST Attack

![Image of Recover RST](image2)

b. Recover RST

![Image of Recover Image](image3)

c. Recover image

![Image ofExtracted Share-1](image4)

d. Extracted Share-1

![Image of Decode QR-code](image5)

e. Decode QR-code

![Image of PSNR and MSE](image6)

f. PSNR and MSE

**Figure 4.11: Rotation-Translation-Scale Attacks**

As shown in the Figure 4.11 the first Figure a shows the Rotation-Translation-Scale attack apply to the watermark image at the value of Rotation $30^\circ$, Translation $10$ & scale $2$. Now shown in b, using surf feature and affine transformation the Rotation-Translation-Scale is recovered near by the attack translation-scale. As shown in c, I applied the affine transformation to recover the watermark image translation-scale. So by applying extraction I got the share-1 as shown in d. at the end by combining the share-1 and share-2 will get the exact QR-code image as shown in e. and calculating the PSNR and MSE value as shown in f.
## 4.6 EXPERIMENTAL ANALYSIS

### Table 4.1: Rotation with Scale

<table>
<thead>
<tr>
<th>Rotation</th>
<th>Scale</th>
<th>PSNR (db)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>63.083</td>
<td>0.028</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>64.043</td>
<td>0.025</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>65.063</td>
<td>0.023</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td>65.053</td>
<td>0.022</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>66.021</td>
<td>0.021</td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>66.081</td>
<td>0.023</td>
</tr>
<tr>
<td>55</td>
<td></td>
<td>64.093</td>
<td>0.022</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>65.023</td>
<td>0.024</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>65.033</td>
<td>0.019</td>
</tr>
<tr>
<td>120</td>
<td></td>
<td>66.071</td>
<td>0.022</td>
</tr>
<tr>
<td>180</td>
<td></td>
<td>64.081</td>
<td>0.024</td>
</tr>
</tbody>
</table>

### Table 4.2: Translation with Rotation

<table>
<thead>
<tr>
<th>Rotation</th>
<th>Translation</th>
<th>PSNR (db)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>62.051</td>
<td>0.032</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>63.072</td>
<td>0.033</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>64.022</td>
<td>0.031</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td>65.064</td>
<td>0.028</td>
</tr>
<tr>
<td>40</td>
<td>10</td>
<td>66.014</td>
<td>0.025</td>
</tr>
<tr>
<td>45</td>
<td></td>
<td>66.032</td>
<td>0.023</td>
</tr>
<tr>
<td>55</td>
<td></td>
<td>64.086</td>
<td>0.025</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>65.068</td>
<td>0.026</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>65.029</td>
<td>0.012</td>
</tr>
<tr>
<td>120</td>
<td></td>
<td>65.032</td>
<td>0.021</td>
</tr>
<tr>
<td>180</td>
<td></td>
<td>64.055</td>
<td>0.023</td>
</tr>
</tbody>
</table>
Figure 4.12: MSE of Rotation with Translation and Scale

Figure 4.13: PSNR of Rotation varying and Translation-10, Scale-2

Table 4.3: Translation varying with Scale-2 attacks

<table>
<thead>
<tr>
<th>Translation</th>
<th>Scale</th>
<th>PSNR (db)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>66.023</td>
<td>0.032</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>66.019</td>
<td>0.033</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>66.015</td>
<td>0.031</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>66.011</td>
<td>0.030</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>66.007</td>
<td>0.031</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>66.003</td>
<td>0.023</td>
</tr>
<tr>
<td>-2</td>
<td></td>
<td>65.999</td>
<td>0.024</td>
</tr>
<tr>
<td>-3</td>
<td></td>
<td>65.995</td>
<td>0.028</td>
</tr>
<tr>
<td>-4</td>
<td></td>
<td>65.991</td>
<td>0.028</td>
</tr>
<tr>
<td>-5</td>
<td></td>
<td>65.987</td>
<td>0.027</td>
</tr>
</tbody>
</table>
Figure 4.14: PSNR of Scale and Translation

Figure 4.15: MSE of Scale and Translation

**Conclusion:** In this chapter, Embedding Process is done by DWT+SVD in Block manner. Surf and Pseudo Zernike moment feature is used to detect rotation and scaling attack. Affine transformation is used to recover watermark image attacks. The analysis of above process is verify using two parameters PSNR and MSE. The values of parameters for Dual RST attacks (Rotation + Scale, Translation + Rotation and Translation + Scale) are show in Table 4.1, Table 4.2 and Table 4.3 respectively. It concludes that the PSNR value increases above 60db and MSE value decreases below 0.20 for color image.